University of New Mexico. Department of Electrical and Computer Engineering.

ECE 331. Second Midterm. March 2017.

1	2	3	T
4.5	5.5	4	14

NAME. Francisco Viramontes

(1) (8 POINTS) Consider the recurrence

$$T(n) = 4T(n/2) + n^2 \lg n.$$

(a) Check that the master method cannot be applied to this recurrence; (3 POINTS) 2

(b) Use a recursion tree to determine a good asymptotic upper bound. What is the height of the tree? What is the number of leaves of the tree? Get a bound for the inside cost and for the fringe cost. (3 POINTS) 1.5

(c) Use the substitution method to solve the recurrence using the bound found in part (b). (2 POINTS). 1

(2) (6 POINTS) Suppose that you are given an array $A[1..n]$ with a max-heap structure. If you run HEAPSORT on the array, obviously you will find the **largest, second largest, third largest and smallest** elements in $\Theta(n \lg n)$ running time. Argue using plain English, however, that you can do better, and find those elements using an algorithm that uses strictly less than n comparisons, for n large.

(3) (6 POINTS) What is the running time of QUICKSORT when all elements of the n-long array A have the same value? When all elements are distinct and are sorted in increasing order? Justify your answers.

Francisco Viramontes

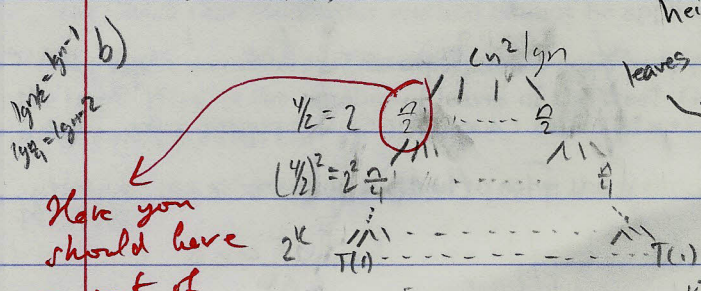
1. $T(n) = 4T(n/2) + n^2 \lg n \sim aT(n/b) + n^2 \lg n$

a) applying the master method: $n^{\log_b a} = n^{\log_2 4} = n^{\lg 4} = n^2$

We cannot determine if $n^2 = \Theta(n^2 \lg n)$, $n^2 = O(n^2 \lg n)$, or $n^2 = \Omega(n^2 \lg n)$ given the criteria of the master method (it is not any of the cases).

You need a more careful analysis of the growth of $n^2 \lg n$ vs. n^2, $n^{2-\epsilon}$, $n^{2+\epsilon}$

b)



Here you should have a cost of $(\frac{n}{2})^2 \lg(\frac{n}{2})$

height $\rightarrow \frac{n}{2^k} = 1 \Rightarrow n = 2^k \Rightarrow k = \lg n$ ✓

leaves $\rightarrow 4^k$ leaves $\Rightarrow 4^{\lg n} \lg n = n^{\lg 4} \lg n$ ✗

$= n^2 \lg n \leftarrow$ Fringe cost

$\lg n \left[1 + (2)^1 + (2)^2 + \dots + (2)^k\right] = \lg n \left[\frac{1-(2)^{k+1}}{1-2}\right] = \lg n \left[1-(2)^k\right] \sim \lg n (2)^k$

$= \lg n \cdot 2^{\lg n} = n^{\lg 2} \lg n = n \lg n \leftarrow$ Inside cost

A good bound based on the calculations above is $n^2 \lg n$

c) $T(n) \leq cn^2 \lg n$

$T(n) \leq 4c\left[\frac{n}{2}\right]^2 \lg(\frac{n}{2}) + n^2 \lg n = 4c(\frac{n^2}{4})[\lg n - 1] + n^2 \lg n$

$= cn^2 \lg n - cn^2 + n^2 \lg n \leq cn^2 \lg n$ ✓

$n^2 \lg n \leq cn^2 \Rightarrow c \geq \lg n$ does not look right

Guess #2 $T(n) \leq cn^2 \lg n - dn \lg n$

$T(n) \leq 4\left[c(\frac{n}{2})^2 \lg(\frac{n}{2}) - d(\frac{n}{2})\lg(\frac{n}{2})\right] + n^2 \lg n = 4\left[c\frac{n^2}{4}[\lg n - \lg 2^1]\right] - d\frac{n}{2}[\lg n - 1] + n^2 \lg n$

$= cn^2 \lg n - cn^2 - d2n \lg n - 2dn + n^2 \lg n \leq cn^2 \lg n - cn^2 - d2n^2 - 2dn^2 + n^2 \lg n$

$-cn^2 - dn^2 - 2dn^2 + n^2 \lg n \leq 0 \Rightarrow n^2 \lg n \leq cn^2 + dn^2 + 2dn^2$

$\lg n \leq c + 3d$ well this is not good...

Guess #3 $T(n) \leq cn^2 \lg n - d \lg n$

$T(n) \leq 4\left[c\frac{n^2}{4}\lg(\frac{n}{2})\right] - d\lg n + n^2 \lg n = cn^2 \lg n - cn^2 - 4d \lg n + 4d + n^2 \lg n$

Good tries, but you have the initial guess wrong...

2. Array A[1...n] w/max heap structre, we can find largest, 2nd, 3rd & smallest elements in $\Theta(n \lg n)$ run time. Argue we can do better $< n$ comp for n

(5-5)

but building a min heap would be $O(n)$ and it might not be strictly linear

Well, we could use Build max heap since it is $\Theta(n)$, for $n-1$ comparasons we can find the maximum of a heap structure and pop of the top of the heap since it will be the largest. We can rinse & repeat for the next two largest elements. For the min we can modify build max heap into build min heap and run $n-1$ comparasons to find the smallest value of a heap length n.

3. The run time of Quick sort if all elements of A would be worst case (SC) senario since we have to consistently make $n-1$ comparisons for n elements. $T(n) = T(\frac{n}{2}) + n$ The run time would be $\Theta(n^2)$.

<1,2,3,4>
<1,2,3>
<1,2>
<1>

If all the elements were in increasing order it would be $\Theta(n)$ because it starts of by making $n-1$ comparasons then pivot is in its final place then does $n-2$ comparasons then it keeps going until it has one comparason left.

$\underbrace{(n-1)+(n-2)+(n-3)+\cdots+1}_{1+2+\cdots+n-1 \;=\; \frac{n(n-1)}{2} \;=\; \Theta(n^2)} \sim \Theta(n)$ ✗